

Docket No.: 60188-694

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of	:	Customer Number: 20277
	:	
Makoto FUJIWARA, et al.	:	Confirmation Number:
	:	
Serial No.:	:	Group Art Unit:
	:	
Filed: October 30, 2003	:	Examiner: Unknown
	:	
For: PROGRAM UPDATE METHOD AND SERVER	:	

**CLAIM OF PRIORITY AND  
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Mail Stop CPD  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

**Japanese Patent Application No. 2002-331992, filed November 15, 2002**

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

  
Michael E. Fogarty  
Registration No. 36,139

600 13<sup>th</sup> Street, N.W.  
Washington, DC 20005-3096  
(202) 756-8000 MEF:tlb  
Facsimile: (202) 756-8087  
**Date: October 30, 2003**

60188-694  
FUJIWARA et al.  
October 30, 2003

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

*McDermott, Will & Emery*

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 2 年 1 1 月 1 5 日

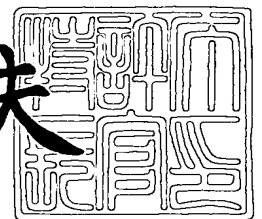
出 願 番 号  
Application Number: 特 願 2 0 0 2 - 3 3 1 9 9 2  
[ST. 10/C]: [ J P 2 0 0 2 - 3 3 1 9 9 2 ]

出 願 人  
Applicant(s): 松 下 電 器 産 業 株 式 会 社

2 0 0 3 年 8 月 1 2 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 6 4 8 4 5

【書類名】 特許願

【整理番号】 5037540113

【提出日】 平成14年11月15日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/00

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 藤原 睦

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 根本 祐輔

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 安井 純一

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 前田 卓治

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 伊藤 孝幸

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 山田 泰司

**【発明者】**

**【住所又は居所】** 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

**【氏名】** 井上 信治

**【特許出願人】**

**【識別番号】** 000005821

**【氏名又は名称】** 松下電器産業株式会社

**【代理人】**

**【識別番号】** 100077931

**【弁理士】**

**【氏名又は名称】** 前田 弘

**【選任した代理人】**

**【識別番号】** 100094134

**【弁理士】**

**【氏名又は名称】** 小山 廣毅

**【選任した代理人】**

**【識別番号】** 100110939

**【弁理士】**

**【氏名又は名称】** 竹内 宏

**【選任した代理人】**

**【識別番号】** 100110940

**【弁理士】**

**【氏名又は名称】** 嶋田 高久

**【選任した代理人】**

**【識別番号】** 100113262

**【弁理士】**

**【氏名又は名称】** 竹内 祐二

## 【選任した代理人】

【識別番号】 100115059

【弁理士】

【氏名又は名称】 今江 克実

## 【選任した代理人】

【識別番号】 100115510

【弁理士】

【氏名又は名称】 手島 勝

## 【選任した代理人】

【識別番号】 100115691

【弁理士】

【氏名又は名称】 藤田 篤史

## 【手数料の表示】

【予納台帳番号】 014409

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0006010

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム更新方法およびサーバ

【特許請求の範囲】

【請求項 1】 L S I と外部メモリとを含むシステムにおいて、前記外部メモリに格納され、前記 L S I 固有の固有鍵で暗号化された固有鍵暗号化プログラムを更新する方法であって、

前記システムが、サーバから送信された、共有鍵で暗号化された共有鍵暗号化プログラムを受信する第 1 ステップと、

前記システムが、受信した前記共有鍵暗号化プログラムを復号することによって、平文プログラムを生成する第 2 ステップと、

前記システムが、前記平文プログラムを、前記固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして前記外部メモリに格納する第 3 ステップとを備えたことを特徴とするプログラム更新方法。

【請求項 2】 請求項 1 記載のプログラム更新方法において、

前記サーバから送信された共有鍵情報を、受信するステップと、

受信した前記共有鍵情報を用いて、平文共有鍵を生成するステップとをさらに備え、

前記第 2 ステップにおいて、前記平文共有鍵を用いて、前記共有鍵暗号化プログラムを復号することを特徴とするプログラム更新方法。

【請求項 3】 請求項 2 記載のプログラム更新方法において、

前記共有鍵情報は、前記平文共有鍵を平文第 1 中間鍵で暗号化した暗号化共有鍵と、前記平文第 1 中間鍵を平文第 2 中間鍵で暗号化した暗号化第 1 中間鍵とを含むものである

ことを特徴とするプログラム更新方法。

【請求項 4】 請求項 1 記載のプログラム更新方法において、

前記 L S I は、固有鍵情報が格納された内部メモリを備えており、

前記システムは、その起動時に、前記内部メモリに格納された固有鍵情報を用いて、平文固有鍵を生成するものであり、

前記第 3 ステップにおいて、前記平文プログラムの再暗号化のために、前記平文固有鍵を用いることを特徴とするプログラム更新方法。

【請求項 5】 請求項 4 記載のプログラム更新方法において、  
前記固有鍵情報は、前記平文固有鍵を平文第 3 中間鍵で暗号化した暗号化固有鍵と、前記平文第 3 中間鍵を平文第 4 中間鍵で暗号化した暗号化第 2 中間鍵とを含むものであることを特徴とするプログラム更新方法。

【請求項 6】 請求項 4 記載のプログラム更新方法において、  
生成された前記平文固有鍵は、前記 L S I 内のレジスタに格納され、前記固有鍵暗号化プログラムを実行する際の、平文プログラムへの復号化のために、用いられることを特徴とするプログラム更新方法。

【請求項 7】 請求項 1 記載のプログラム更新方法において、  
前記 L S I は、ブートプログラムが格納されたブート R O M を備えており、  
前記外部メモリは、サーバとの送受信を実行するための取得プログラムが格納されており、  
前記システムは、前記共有鍵暗号化プログラムの受信を、前記外部メモリに格納された取得プログラムによって実行し、受信後の更新処理を、前記ブート R O M に格納されたブートプログラムによって制御することを特徴とするプログラム更新方法。

【請求項 8】 請求項 1 記載のプログラム更新方法において、  
前記サーバから送信された、前記平文プログラムのハッシュ値を受信するステップを備え、  
前記第 2 ステップにおいて、復号した平文プログラムについて、受信したハッシュ値を用いてハッシュ検証を行うことを特徴とするプログラム更新方法。

【請求項 9】 L S I を含むシステムにおけるプログラム更新のために、動作するサーバであって、

前記システムから、前記 L S I の I D と、更新対象プログラムの識別情報であるアプリ I D とを受信する第 1 ステップと、

アプリ I D と L S I I D との対応関係を示す第 1 のテーブルを参照して、前記更新対象プログラムを前記システムに送信するか否かを決定する第 2 ステップと

、  
前記第 2 ステップで送信すると決定したとき、前記システムへ、前記更新対象プログラムを共有鍵で暗号化した共有鍵暗号化プログラム、および前記共有鍵の基になる共有鍵情報を送信する第 3 ステップとを実行することを特徴とするサーバ。

【請求項 1 0】 請求項 9 記載のサーバにおいて、

前記システムから、前記更新対象プログラムの実行に必要なアプリ固有情報を要求する信号を受信する第 4 ステップと、

アプリ固有情報の送信履歴と L S I I D との対応関係を示す第 2 のテーブルを参照して、前記第 4 のステップで要求されたアプリ固有情報を送信するか否かを決定する第 5 のステップとを実行することを特徴とするサーバ。

【請求項 1 1】 請求項 9 記載のサーバにおいて、

前記共有鍵情報は、平文共有鍵を平文第 1 中間鍵で暗号化した暗号化共有鍵と、前記平文第 1 中間鍵を平文第 2 中間鍵で暗号化した暗号化第 1 中間鍵とを含むものである

ことを特徴とするサーバ。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、鍵実装されたシステムやこれに用いる L S I において、セキュリティを保ちつつ、プログラムを更新する技術に関する。

【 0 0 0 2 】

【従来の技術】

従来、L S I を動作させるプログラムを不正な処理から守るために、予め定ま



ったメーカー鍵で暗号化したプログラムをメモリに記憶しておき、復号化して実行することが行われている。しかしながら、このようなシステムでは共通のメーカー鍵で暗号化されたプログラムを実行する L S I が大量にあるため、たとえ 1 個の製品からでもメーカー鍵の情報が不正に漏洩すると、大量の製品においてプログラムを改ざん可能になってしまい、したがって、セキュリティを高めることができないという課題がある。

#### 【 0 0 0 3 】

この課題を解決するために、L S I を動作させるプログラムを L S I 毎に固有の固有鍵で暗号化し、製品としては固有鍵で暗号化されたプログラムのみを実行可能とする手法がある（特願 2 0 0 2 - 2 1 5 0 9 6、特願 2 0 0 2 - 2 5 8 4 8 1 参照）。この手法を用いることにより、1 個の製品で鍵情報が不正に漏洩したとしても、その他の製品への影響はないので、セキュリティを高めることができる。また、この手法の前提として、鍵を二重に暗号化する手法がある（特願 2 0 0 1 - 2 8 6 8 8 1 参照。）。

#### 【 0 0 0 4 】

なお、ここで挙げた特許出願はいずれも未だ出願公開されておらず、このため、記載すべき先行技術文献情報はない。

#### 【 0 0 0 5 】

##### 【発明が解決しようとする課題】

一般に、製品となっている L S I に実装されたプログラムのアップデート（更新）は、S S L 接続によって通信路を安全に確保し、平文プログラム、またはメーカー鍵で暗号化されたプログラムをサーバから L S I へ送信することによって行われている。しかしながら、この手法では、通信路を不正にアクセスされると、大量の製品で実行可能なプログラムが不正に入手されてしまうので、プログラム更新におけるセキュリティを高めることができない。

#### 【 0 0 0 6 】

これを解決するために、上述の手法を用いた場合、L S I は固有鍵で暗号化されたプログラムのみを実行するので、平文プログラムやメーカー鍵で暗号化されたプログラムを送信しても、そのままでは実行することができない。

**【 0 0 0 7 】**

また、L S I 毎に異なる鍵で暗号化されたプログラムをサーバ側で準備し、L S I 毎に鍵情報を管理した上で、サーバから L S I へ、L S I 毎に異なる鍵で暗号化されたプログラムを送信する方法も考えられるが、この方法では、莫大な手間とコストを必要とするため、現実的ではない。

**【 0 0 0 8 】**

前記の問題に鑑み、本発明は、L S I 固有の固有鍵で暗号化されたプログラムを実行可能な L S I について、高いセキュリティを保ちつつ、プログラムを更新する方法を提供することを課題とする。

**【 0 0 0 9 】****【課題を解決するための手段】**

前記の課題を解決するために、本発明が講じた解決手段は、L S I と外部メモリとを含むシステムにおいて、前記外部メモリに格納され、前記 L S I 固有の固有鍵で暗号化された固有鍵暗号化プログラムを更新する方法として、前記システムが、サーバから送信された共有鍵で暗号化された共有鍵暗号化プログラムを受信する第 1 ステップと、前記システムが、受信した前記共有鍵暗号化プログラムを復号することによって平文プログラムを生成する第 2 ステップと、前記システムが、前記平文プログラムを前記固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして前記外部メモリに格納する第 3 ステップとを備えたものである。

**【 0 0 1 0 】**

前記本発明に係るプログラム更新方法において、前記サーバから送信された共有鍵情報を受信するステップと、受信した前記共有鍵情報を用いて平文共有鍵を生成するステップとをさらに備え、前記第 2 ステップにおいて、前記平文共有鍵を用いて前記共有鍵暗号化プログラムを復号するのが好ましい。

**【 0 0 1 1 】**

さらに、前記共有鍵情報は、前記平文共有鍵を平文第 1 中間鍵で暗号化した暗号化共有鍵と、前記平文第 1 中間鍵を平文第 2 中間鍵で暗号化した暗号化第 1 中間鍵とを含むのが好ましい。

**【 0 0 1 2 】**

また、前記本発明に係るプログラム更新方法において、前記 L S I は、固有鍵情報が格納された内部メモリを備えており、前記システムは、その起動時に、前記内部メモリに格納された固有鍵情報を用いて平文固有鍵を生成するものとし、前記第 3 ステップにおいて、前記平文プログラムの再暗号化のために前記平文固有鍵を用いるものとする。

#### 【 0 0 1 3 】

さらに、前記固有鍵情報は、前記平文固有鍵を平文第 3 中間鍵で暗号化した暗号化固有鍵と、前記平文第 3 中間鍵を平文第 4 中間鍵で暗号化した暗号化第 2 中間鍵とを含むのが好ましい。あるいは、生成された前記平文固有鍵は、前記 L S I 内のレジスタに格納され、前記固有鍵暗号化プログラムを実行する際の平文プログラムへの復号化のために用いられるのが好ましい。

#### 【 0 0 1 4 】

また、前記本発明に係るプログラム更新方法において、前記 L S I は、ブートプログラムが格納されたブート R O M を備えており、前記外部メモリは、サーバとの送受信を実行するための取得プログラムが格納されており、前記システムは、前記共有鍵暗号化プログラムの受信を、前記外部メモリに格納された取得プログラムによって実行し、受信後の更新処理を、前記ブート R O M に格納されたブートプログラムによって制御するのが好ましい。

#### 【 0 0 1 5 】

また、前記本発明に係るプログラム更新方法において、前記サーバから送信された前記平文プログラムのハッシュ値を受信するステップを備え、前記第 2 ステップにおいて、復号した平文プログラムについて、受信したハッシュ値を用いてハッシュ検証を行うのが好ましい。

#### 【 0 0 1 6 】

また、本発明が講じた解決手段は、L S I を含むシステムにおけるプログラム更新のために動作するサーバとして、前記システムから前記 L S I の I D と更新対象プログラムの識別情報であるアプリ I D とを受信する第 1 ステップと、アプリ I D と L S I I D との対応関係を示す第 1 のテーブルを参照して、前記更新対象プログラムを前記システムに送信するか否かを決定する第 2 ステップと、前記

第2ステップで送信すると決定したとき、前記システムへ、前記更新対象プログラムを共有鍵で暗号化した共有鍵暗号化プログラム、および前記共有鍵の基になる共有鍵情報を送信する第3ステップとを実行するものである。

#### 【0017】

そして、前記本発明に係るサーバにおいて、前記システムから、前記更新対象プログラムの実行に必要なアプリ固有情報を要求する信号を受信する第4ステップと、アプリ固有情報の送信履歴とLSIIDとの対応関係を示す第2のテーブルを参照して、前記第4のステップで要求されたアプリ固有情報を送信するか否かを決定する第5のステップとを実行するのが好ましい。

#### 【0018】

また、前記本発明に係るサーバにおいて、前記共有鍵情報は、平文共有鍵を平文第1中間鍵で暗号化した暗号化共有鍵と、前記平文第1中間鍵を平文第2中間鍵で暗号化した暗号化第1中間鍵とを含むのが好ましい。

#### 【0019】

##### 【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照して説明する。なお、以下の説明では、X（鍵またはプログラム）を鍵Yを用いて暗号化して得た、暗号化された鍵またはプログラムのことを、Enc（X，Y）と表すものとする。

#### 【0020】

図1は本実施形態に係る半導体装置としてのセキュアLSIの内部構成を示すブロック図である。図1において、セキュアLSI1は外部バス120を介して、外部メモリ（フラッシュメモリ）100や外部ツール110などと接続可能に構成されている。また、モードIDを与えることによって、その動作モードを設定することが可能になっている。

#### 【0021】

本実施形態に関わる主な構成要素について、簡単に説明する。

#### 【0022】

まず、セキュアLSI1は、書き換え不可領域11を含むセキュアメモリ（セキュアFlash）10を備えている。この書き換え不可領域11には、書き換

え不可領域書き込みフラグ 1 2 が設けられている。書き換え不可領域書き込みフラグ 1 2 は、モード I D が一度セキュアメモリ 1 0 に書き込まれると、そのフラグ値が“可”から“済”になり、それ以降の書き換え不可領域への書き込みが不能になる。なお、本実施形態では、セキュアメモリ 1 0 および外部メモリ 1 0 0 はフラッシュメモリによって構成されているが、もちろんこれに限定されるものではなく、不揮発性のメモリであればどのようなものであってもかまわない。

### 【0 0 2 3】

また、暗号化部 2 はプログラムの暗号化や復号化を行うものであり、秘密鍵演算処理部 2 0 と、鍵生成・更新シーケンサ 3 0 とを備えている。秘密鍵演算処理部 2 0 は各種の鍵、およびプログラム暗号種を格納するレジスタ（プログラム共有鍵格納レジスタ 2 1、プログラム固有鍵格納レジスタ 2 2、暗号鍵格納レジスタ 2 3 等）を備えており、プログラムの暗号化処理又は復号化処理を含む複数のシーケンスを実行可能である。鍵生成・更新シーケンサ 3 0 は秘密鍵演算処理部 2 0 が実行可能な各シーケンスについて実行の諾否を判断し、実行が許されないと判断したシーケンスについて秘密鍵演算処理部 2 0 の動作を禁止する。鍵生成更新シーケンサ 3 0 はモード I D 格納レジスタ 3 1 を有しており、このモード I D 格納レジスタ 3 1 に格納されているモード I D に応じて、各シーケンスの実行の諾否を判断する。また、鍵またはプログラムがどのようなアルゴリズムや鍵長で暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ 3 2、およびプログラム暗号種を記憶する記憶部 3 3 を備えている。

### 【0 0 2 4】

モードシーケンサ 4 0 も、モード I D 格納レジスタ 4 1 を備えており、モード I D 格納レジスタ 4 1 に格納されているモード I D と、ジャンパ 4 3 の値に応じて、外部ホストインターフェース（I/F）5 0 の動作、すなわち、外部メモリ 1 0 0 に格納されたプログラムやデータをどの I/F を介して読み込むか、を制御する。これにより、外部メモリ 1 0 0 に格納された平文プログラムが実行できるか否かを制御することができる。さらに、モードシーケンサ 4 0 は、鍵がどの手法によって暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ 4 2 を備えている。

**【0025】**

外部ホスト I/F 50 は、モードシーケンサ 40 の制御に従って、プログラム処理部 51 が有する暗号化用スルー部 52、実行用スルー部 53 およびプログラム復号用暗号エンジン 54、並びに、データ処理部 55 が有するスルー部 56 およびコンテンツ暗号・復号用暗号エンジン 57 のうちのいずれかを介して、外部メモリ 100 や外部ツール 110 との間でプログラムやデータの入出力を行う。またプログラム復号用暗号エンジン 54 は、プログラムの復号に用いるプログラム固有鍵を格納するためのプログラム固有鍵可能レジスタ 58 を備えている。

**【0026】**

ここで、後述する鍵生成モードと商品動作モードにおいては、実行用スルー部 53 を介してプログラムを取り込むことが出来ないように構成されている。すなわち、セキュア LSI 1 は後述する鍵生成モードと商品動作モードにおいては、固有鍵で暗号化されたプログラム以外へは動作を遷移しないように構成されている。

**【0027】**

ブート ROM 60 は、セキュア LSI 1 の起動動作を制御するブートプログラムを格納している。HASH 演算部 70 は、セキュア LSI 1 に読み込まれたプログラムについてその正当性を検証するために、HASH 値を演算する。

**【0028】**

また、外部メモリ 100 には、プログラムやコンテンツが格納されている。外部ツール 110 には、セキュア LSI 1 の最初の起動時にセキュアメモリ 10 に格納する各種の初期値が格納されている。この初期値の種類は、設定される動作モードに応じて、異なったものになる。

**【0029】**

図 2 は図 1 のセキュア LSI 1 を用いた開発および製品化の全体の流れを表す図である。図 2 に示すように、セキュア LSI 1 は、アドミニストレータモード（モード ID：00）、鍵生成モード（モード ID：01）、開発モード（モード ID：10）および商品動作モード（モード ID：11）の 4 種類の動作モードで、動作する。

**【 0 0 3 0 】**

まず、アドミニストレータモードに設定されたセキュア L S I 1 は、管理者用 L S I として、動作する。管理者用 L S I では、鍵生成プログラムが開発され（ P A 1 ）、また、その鍵生成プログラムが任意の鍵生成鍵を用いて暗号化される（ P A 2 ）。

**【 0 0 3 1 】**

鍵生成モードに設定されたセキュア L S I 1 は、鍵生成用 L S I として、動作する。鍵生成用 L S I では、管理者用 L S I において生成された、暗号化された鍵生成プログラムが実装され（ P B 1 ）、この鍵生成プログラムを実行することによって、各種の鍵が生成される（ P B 2 ）。

**【 0 0 3 2 】**

開発モードに設定されたセキュア L S I 1 は、開発用 L S I として、動作する。開発用 L S I では、実際の製品で実行されるアプリケーション用プログラムが開発される（ P C 1 ）。そして、このアプリケーション用プログラムが、プログラム共有鍵を用いて暗号化される（ P C 2 ）。

**【 0 0 3 3 】**

商品動作モードに設定されたセキュア L S I 1 は、実際の商品 L S I として、動作する。商品 L S I では、開発用 L S I において生成された、プログラム共有鍵で暗号化されたアプリケーション用プログラムが実装され、その内部で、プログラム固有鍵で暗号化されたアプリケーション用プログラムに、変換される（ P D 1 ）。その後、通常の商品 L S I として動作する（ P D 2 ）。なお、この変換処理は、開発用 L S I でも、アプリケーション用プログラムのデバッグのために、実行可能になっている（ P C 3 ）。

**【 0 0 3 4 】**

以下、上記のように構成されたセキュア L S I 1 の商品動作モードにおける通常動作およびセキュアアップデート動作の詳細について、フローチャートおよびデータフローを参照して、説明する。

**【 0 0 3 5 】**

図 3 はブートプログラムの全体的な処理の流れを示すフローチャートである。

セキュア L S I 1 に電源が投入されると、ブート R O M 6 0 に格納されたブートプログラムが C P U 6 5 によって実行される。図 3 に示すように、まず、各ハードウェアを初期化する ( S Z 0 ) 。そして、外部ツール 1 1 0 からさまざまな初期値を読み込み、セキュアメモリ 1 0 に設定する ( S Z 1 ) 。

#### 【0036】

図 4 は初期値設定処理 S Z 1 のフローチャートである。まず、ジャンパ 4 4 で、セキュアメモリ 1 0 が L S I 内に実装されているか否かの判定を行う ( S Z 1 1 ) 。次に、書き換え不可領域書き込みフラグ 1 2 が “済” であるか否かを判定し ( S Z 1 2 ) 、 “済” であるときは ( Y e s ) すでにセキュアメモリ 1 0 に初期値が設定されているので、処理 S Z 1 を終了する。書き換え不可領域書き込みフラグ 1 2 が “可” であるときは ( N o ) 、セキュアメモリ 1 0 に初期値を書き込んでいく ( S Z 1 3 ~ S Z 1 8 ) 。モード I D に加えて、暗号化されたプログラム固有鍵、アドレス管理情報、データ固有鍵をセキュアメモリ 1 0 の書き換え不可領域 1 1 に書き込む。なお、最初の判定の結果、セキュアメモリ 1 0 が L S I の外部にあると判定されたときは ( S Z 1 4 で N o ) 、モード I D は商品動作モードを表す値に上書きされる ( S Z 1 5 ) 。これにより、セキュアメモリ 1 0 が L S I パッケージ外にあるような不正な製品は、商品動作モードでしか動作できない。

#### 【0037】

次に、書き込み不可領域書き込みフラグ 1 2 を “済” にセットする ( S Z 1 9 ) 。これによって、以後の書き換え不可領域 1 1 の書き換えはできなくなる。さらに、通常領域 1 3 , 1 4 に暗号種別識別子および実装モードフラグを書き込む ( S Z 1 A ) 。そして、モード I D がアドミニストレータモード以外のモードを示すときは ( S Z 1 B で N o ) 、これらに加えて、暗号化された共有鍵／鍵生成鍵も通常領域 1 3 , 1 4 に書き込む ( S Z 1 C ) 。

#### 【0038】

その後、前処理 S Z 2 を実行する。ここでは、セキュアメモリ 1 0 の書き込み不可領域 1 1 に設定されたモード I D が、鍵生成・更新シーケンサ 3 0 のモード I D 格納レジスタ 3 1 と、モードシーケンサ 4 0 のモード I D 格納レジスタ 4 1



とに設定される。また、セキュアメモリ 1 0 の第 1 の通常領域 1 3 に設定された暗号種別識別子が、鍵生成・更新シーケンサ 3 0 の暗号種別識別子格納レジスタ 3 2 と、モードシーケンサ 4 0 の暗号種別識別子格納レジスタ 4 2 とに設定される。さらに、セキュアメモリ 1 0 の書き換え不可領域 1 1 に格納されたアドレス管理情報が、MEMC 8 0 の暗号アドレス区分格納レジスタ 8 1 に設定される。ここまでの動作は、図 2 における初期値設定フェーズ P A 0 , P B 0 , P C 0 , P D 0 に対応している。

#### 【 0 0 3 9 】

その後は、モード I D の値に応じて、それぞれのモードにおける動作を行う（S Z 3）。このようにモード I D の値に応じて、セキュア L S I で行う動作を制限することによりプログラムの秘匿性を高めている。

#### 【 0 0 4 0 】

次に通常の商品動作（通常ブート処理）について詳細に説明する。

#### 【 0 0 4 1 】

モード I D が「1 1」のとき、セキュア L S I 1 は商品動作モードになり、実装モードフラグの値に応じて（S D 0）、プログラム実装処理 S D 1、または通常ブート処理 S D 2 を実行する。

#### 【 0 0 4 2 】

図 5 はプログラム実装処理 S D 1 のフローチャート、図 6、7 はデータフローである。プログラム実装処理 S D 1 においては、セキュアメモリ 1 0 に格納された固有鍵情報を用いてプログラム固有鍵を（S D 1 1、S D 1 2）、共有鍵情報を用いてプログラム共有鍵を復号し（S D 1 3、S D 1 4）、復号されたプログラム共有鍵とプログラム固有鍵を用いて外部メモリ 1 0 0 に格納されたプログラム E n c（プログラム、プログラム共有鍵）を E n c（プログラム、プログラム固有鍵）へと変換する（S D 1 5－S D 1 7）。その後、プログラムの正当性の検証を行い（S D 1 8）、正当であるならば実装モードフラグを O F F に設定する（S D 1 9）。これにより次回の起動時からはプログラム実装処理 S D 1 は行われない。そして最後にセキュアメモリ 1 0 に格納されたプログラム共有鍵と、外部メモリ 1 0 0 に格納されたプログラム E n c（プログラム、プログラム共有

鍵)は削除される(SD1A、SD1B)。

#### 【0043】

図8は通常ブート処理SD2のフローチャート、図9、10はデータフローである。通常ブート処理SD2においては、まず、内部メモリとしてのセキュアメモリ10の書き込み不可領域11に格納された、固有鍵情報としての暗号化されたプログラム固有鍵、すなわち、暗号化固有鍵Enc(プログラム固有鍵(平文)、MK0(平文第3中間鍵))および暗号化第2中間鍵Enc(MK0, CK0(平文第4中間鍵))を秘密鍵演算処理部20の暗号鍵格納レジスタに設定する(SD21)。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ30に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る(SD22)。得られたプログラム固有鍵は秘密鍵演算処理部20のプログラム固有鍵格納レジスタ22と、外部ホストI/F50のプログラム復号用暗号エンジン54のプログラム固有鍵格納レジスタ58に設定する(SD23)。

#### 【0044】

その後、セキュアメモリ10の書き込み不可領域11に格納されているデータ固有IDを秘密鍵演算処理部20の固有ID格納レジスタに設定する(SD24)。また、CPU65によって乱数を生成し、秘密鍵演算処理部20の乱数格納レジスタに設定する(SD25)。そして、秘密鍵演算処理部20によって、データ固有IDと乱数からデータ固有鍵を生成する(SD26)。コンテンツの再生にはデータ固有鍵を用いて行う。データ固有鍵は乱数を用いて生成されるため、起動毎に異なることとなり、コンテンツ再生の安全性が高まる。

#### 【0045】

その後、外部メモリ100に格納されていた、プログラム固有鍵で暗号化されたプログラムEnc(プログラム, プログラム固有鍵)を、外部ホストI/F50が有するプログラム処理部51のプログラム復号用暗号エンジン54を介して復号し、HASH演算部70に取り込み、HASH値を演算する(SD27)。復号に用いられる鍵としては外部ホストI/Fのプログラム固有鍵格納レジスタ58に格納されたプログラム固有鍵が用いられる。そして、この演算したHAS

H 値と、セキュアメモリ 1 0 の通常領域 1 3 に格納されていた H A S H 値とを比較し、プログラムが改ざんされていないかどうかをチェックする（S D 2 8）。H A S H 値が一致していたとき（S D 2 9 で N o）、外部メモリ 1 0 0 に格納されていたプログラム E n c（プログラム、プログラム固有鍵）に処理を遷移し、アプリケーションを実行する（S D 2 A）。一方、H A S H 値が一致していないとき（S D 2 9 で Y e s）は、何らかの不正が行われたものと推定して、不正アクセス時制御による処理を実行する（S D 2 B）。

#### 【 0 0 4 6 】

ここで、上述のように商品として動作するセキュア L S I について、プログラムの更新を行う処理について、図面を参照しながら説明する。図 1 1 はプログラム更新の際に、サーバ 3 とセキュア L S I 1 を含むシステムとの間におけるデータのやり取りを示すフローである。

#### 【 0 0 4 7 】

図 1 1 に示すように、まず、セキュア L S I 1 がプログラム更新処理を起動すると、サーバ 3 はシステムからセキュア L S I 1 の I D を受信して I D 認証を行い、認証した場合は、セキュア L S I 1 と S S L 接続する（U D 1）。これにより、サーバ 3 とセキュア L S I 1 を含むシステムとの間の通信路の安全性が一応確保される。

#### 【 0 0 4 8 】

通信路が確保されると、システムは、更新対象プログラムの識別情報であるアプリ I D をサーバ 3 へ送信する（U D 2）。サーバ 3 は、更新可能なプログラムのアプリ I D と、プログラムを動作させて良い L S I の I D との対応関係を示す第 1 のテーブル 4 を管理しており、この第 1 のテーブル 4 に基づいて、プログラムを送信してよいか否かを判断する。セキュア L S I 1 の I D と、更新を要求されたプログラムのアプリ I D との対応が確認されると、サーバ 3 は、この更新対象プログラムの送信を開始する。

#### 【 0 0 4 9 】

まず、サーバ 3 から、更新対象プログラムの付加情報をセキュア L S I 1 へ送信する（U D 3）。ここでの付加情報は、セキュア L S I 1 側でプログラムを更

新してよいか否かを認証させるための署名、更新対象プログラムのサイズ、および、更新対象プログラムのハッシュ値（平文での値）等を含む。セキュア L S I 1 は、付加情報として送信された署名を用いて認証を行い、また、送信されたプログラムサイズを基にして、外部メモリ 1 0 0 に更新可能な空き領域があるか否かを判断する。そして、更新が可能であると判断した場合は、サーバ 3 に対して、共有鍵情報を送信するよう要求する（U D 4）。

#### 【 0 0 5 0 】

サーバ 3 は、要求を受信すると、共有鍵情報として、暗号化共有鍵 E n c （プログラム共有鍵（平文），MK 1 （平文第 1 中間鍵））および暗号化第 1 中間鍵 E n c （MK 1，CK 1 （平文第 2 中間鍵））をセキュア L S I 1 へ送信する（U D 5）。セキュア L S I 1 は、共有鍵情報を用いてプログラム共有鍵を復号し、復号された状態でハッシュ演算を行い、正当性の検証をする。プログラム共有鍵を正常に復号すると、システムはサーバ 3 に対して、共有鍵暗号化プログラムを送信するよう要求する（U D 6）。サーバ 3 は、要求を受信すると、プログラム E n c （プログラム、プログラム共有鍵）をシステムへ送信する（U D 7）。セキュア L S I 1 は、E n c （プログラム、プログラム共有鍵）を E n c （プログラム、プログラム固有鍵）へと変換する。さらに、変換された E n c （プログラム、プログラム固有鍵）を平文プログラムに復号し、ハッシュ演算して、先に付加情報として受信したハッシュ値との比較によって、正当性を検証する。ここでの処理については、後で詳細に説明する。

#### 【 0 0 5 1 】

共有鍵暗号化プログラムが固有鍵暗号化プログラムに正常に変換できたとき、セキュア L S I 1 を含むシステムはサーバ 3 へ、アプリ固有情報を送信するよう要求する（U D 8）。アプリ固有情報はプログラムの実行に必要な情報を含んでおり、アプリ固有情報がないと、セキュア L S I 1 は更新したプログラムを実行できない。サーバ 3 は、アプリ固有情報の送信履歴と L S I の I D との対応関係を示す第 2 のテーブル 5 も管理しており、同一のセキュア L S I には複数のアプリ固有情報を送信しないようにする。よって、同一のセキュア L S I は複数回、同一のプログラムを更新することができない。

**【0052】**

サーバ3は、アプリ固有情報を送信してよいと判断したとき、アプリ固有情報をセキュアLSI1を含むシステムへ送信し（UD9）、セキュアLSI1がこれをハッシュ演算して正当性を検証すると、プログラム更新が終了し、通信は切断される（UD10）。

**【0053】**

なお、本発明におけるサーバ3とセキュアLSI1を含むシステムとの間におけるデータのやり取りは上述のフローに限定されるものではない。例えば、サーバ3は、必ずしも、第2のテーブルを管理し、同一のセキュアLSI1に複数のアプリ固有情報を送信しないようにする必要はない。しかしながら、同一のセキュアLSI1に複数回、同一のプログラムを送信しないようにすることにより、プログラムの秘匿性はより高まる。

**【0054】**

また、付加情報、共有鍵情報、共有鍵暗号化プログラムは、必ずしも、サーバ3からセキュアLSI1へと別々に送信する必要はなく、その一部、または全部をまとめたプログラムパッケージとして一度に送信してもよい。

**【0055】**

セキュアLSI1における、共有鍵暗号化プログラムEnc（プログラム、プログラム共有鍵）から固有鍵暗号化プログラムEnc（プログラム、プログラム固有鍵）への変換について、図面を参照しながら詳述する。図12は外部メモリ100に格納された、プログラム更新に係わるプログラムの構成を示す図である。

**【0056】**

図12に示すように、外部メモリ100には、固有鍵でそれぞれ暗号化された、暗号化制御プログラム200（Enc（制御プログラム、プログラム固有鍵））、および暗号化アプリケーションプログラム210（Enc（アプリケーションプログラム、プログラム固有鍵））が格納されている。

**【0057】**

暗号化制御プログラム200はアプリケーション起動部201およびプログラ

ム更新制御部 2 0 5 を備え、プログラム更新制御部 2 0 5 は共有鍵復号部 2 0 6、プログラム固有鍵暗号化処理部 2 0 7 およびプログラム更新成否判定部 2 0 8 を備えている。

#### 【0 0 5 8】

アプリケーション起動部 2 0 1 はブート ROM 6 0 に格納されたブートプログラムからの指示を受けて、暗号化アプリケーションプログラム 2 1 0 を起動する。共有鍵復号部 2 0 6 はサーバ 3 から送信された共有鍵情報を基にして、鍵生成・更新シーケンサ 3 0 を用いて、プログラム共有鍵を復号する。プログラム固有鍵暗号化処理部 2 0 7 は鍵生成・更新シーケンサ 3 0 を用いて、共有鍵暗号化プログラム E n c（プログラム、プログラム共有鍵）から固有鍵暗号化プログラム E n c（プログラム、プログラム固有鍵）への変換を行う。プログラム更新成否判定部 2 0 8 は、固有鍵暗号化プログラム E n c（プログラム、プログラム固有鍵）を平文プログラムに復号し、ハッシュ検証によってプログラム更新の成否を判定する。そして、プログラム更新が成功したときは、古いプログラムを削除し、プログラム格納先・サイズなどの情報をセキュアメモリ 1 0 に格納する。

#### 【0 0 5 9】

暗号化アプリケーションプログラム 2 1 0 は、通常の実行プログラムである通常動作部 2 1 1 の他、サーバや記録媒体から新しいアプリケーションプログラムを取得するためのプログラム取得部 2 1 2 もアプリケーションプログラムとして備えている。プログラムの更新は、これらのような外部メモリ 1 0 に格納されたプログラムを用いて行われる。

#### 【0 0 6 0】

図 1 3 は共有鍵暗号化プログラムから固有鍵暗号化プログラムへの変換処理を含む、プログラム更新処理を示すフローチャートである。

#### 【0 0 6 1】

アプリケーションプログラムの実行中（S X 1）に、ユーザの操作等の外部要因によってプログラムの更新が要求されると、システムはこれを検知し、プログラム更新を開始するために、通常動作部 2 1 1 が暗号化アプリケーションプログラム 2 1 0 の取得プログラムとしてのプログラム取得部 2 1 2 を起動する（S X

2)。

#### 【0 0 6 2】

プログラム取得部 2 1 2 はサーバ 3 と通信して、認証や、共有鍵情報・プログラムの取得を行う (S X 3)。サーバ 3 から共有鍵情報を取得すると、共有鍵復号部 2 0 6 はプログラム共有鍵を復号する (S X 4, S X 5)。すなわち、共有鍵情報としての暗号化されたプログラム共有鍵 E n c (プログラム共有鍵, M K 2)、E n c (M K 2, C K) を秘密鍵演算処理部 2 0 の暗号鍵格納レジスタ 2 3 に設定し、この暗号化されたプログラム共有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム共有鍵を得る。得られたプログラム共有鍵は秘密鍵演算処理部 2 0 のプログラム共有鍵格納レジスタ 2 1 に格納される。

#### 【0 0 6 3】

次に、プログラム固有鍵暗号化処理部 2 0 7 が、共有鍵暗号化プログラムから固有鍵暗号化プログラムへの変換を行う。すなわち、サーバ 3 から送信され外部メモリ 1 0 0 に格納されていたプログラム E n c (プログラム, プログラム共有鍵) を、外部ホスト I / F 5 0 が有するプログラム処理部 5 1 の暗号化用スルー部 5 2 を介して、秘密鍵演算処理部 2 0 に取り込む (S X 6')。そして、取り込んだプログラムを、プログラム共有鍵格納レジスタ 2 1 に格納されたプログラム共有鍵で復号した後、プログラム固有鍵格納レジスタ 2 2 に格納されたプログラム固有鍵で暗号化し、プログラム E n c (プログラム, プログラム固有鍵) を得る。なお、上述したように、プログラム固有鍵はシステムの起動時にすでに復号されており、秘密鍵演算処理部 2 0 のプログラム固有鍵格納レジスタ 2 2 に格納されている。

#### 【0 0 6 4】

最後に、プログラム更新成否判定部 2 0 8 が、プログラム更新の成否を判定する。すなわち、E n c (プログラム, プログラム固有鍵) を外部メモリ 1 0 0 に書き込んだ (S X 8) 後、外部ホスト I / F 5 0 が有するプログラム処理部 5 1 のプログラム復号用暗号エンジン 5 3 を用いて、復号して取り込み (S X 9)、平文状態でのハッシュ値を演算する (S X 1 0)。演算されたハッシュ値は、プ

プログラム取得部 212 が暗号化プログラムとともに取得したハッシュ値と比較され、この比較によって更新の成否が判定される (S X 1 1)。更新が成功したときは、古いプログラムを消去する (S X 1 2) 一方、更新が失敗したときは、送信されたプログラムを消去する (S X 1 3)。そしてプログラム格納先、サイズなどの情報をセキュアメモリ 10 に書き込み (S X 1 4)、更新処理が終了する。

#### 【0065】

上述のプログラム更新方法を用いることによって、サーバからプログラム共有鍵暗号化プログラムを送信すると、セキュア L S I において、暗号化する鍵がプログラム共有鍵からプログラム固有鍵に変換されて、システムに実装される。このため、たとえサーバからセキュア L S I への通信路が不正にアクセスされプログラム共有鍵暗号化プログラムが盗み出されたとしても、このプログラムによってセキュア L S I を動作させることは、できない。また更新の結果、ユーザの持つ各製品では、互いに異なる固有鍵によって暗号化されたプログラムが実装されることになり、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数に限定されることになり、従来よりもセキュリティが高まる。

#### 【0066】

なお、本実施形態において、共有鍵情報をサーバから取得しているが、これは、商品動作モード「11」におけるプログラム実装 (S D 1) の最後において、復号されたプログラム共有鍵およびセキュアメモリ 10 上の共有鍵情報を削除しているためであり、これらを削除しない場合は、共有鍵情報はサーバから取得する必要はなく、セキュアメモリ 10 から読み出して復号すればよい。

#### 【0067】

また、本実施形態では、外部要因によってプログラムの更新の開始が指示され、通常サブ 211 がプログラム取得部 212 を起動し、プログラムの取得後はブートプログラムによって各処理を指示するものとしたが、本発明はこれに限られるものではない。例えば、ブートプログラムがプログラム取得部 212 を起動する構成にすることによって、さらにセキュリティを高めることができる。

#### 【0068】



また、プログラム固有鍵は必ずしも製品毎に固有である必要はなく、品種ごとまたは複数個毎に同一であっても良い。1個の製品において暗号を破られたときに被害を少なくするのが本願発明のねらいであり、同一の鍵で暗号化されたプログラムを有するLSIの数を少しでも減らすだけで効果は十分に発揮できる。さらに言えば、プログラム固有鍵は全て共通であったとしても、通信路が破られ共有鍵暗号化プログラムが盗み出されたとしてもそのままではセキュアLSIで動作できないので、鍵を共有鍵から固有鍵に書き換えるだけでも、効果は発揮できる。

#### 【0069】

##### 【発明の効果】

以上のように本発明によると、プログラムをLSI毎の固有鍵で再暗号化して実行する秘匿性の高いセキュアLSIにおいても、サーバから同一のプログラムを送信するだけで、プログラムの更新をすること可能となる。

#### 【0070】

また、サーバからセキュアLSIへの通信路が不正にアクセスされて共有鍵暗号化プログラムが盗み出されたとしても、そのプログラムではセキュアLSIを動作させることができないので、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数が限定されることになり、従来よりもセキュリティが高まる。

#### 【0071】

さらに、サーバから受信した共有鍵やプログラムの正当性を平文状態のハッシュ値を用いて行うので、通信路における暗号化状態でのハッシュ値で行うよりもハッシュ値の改ざんが行いにくくなり、セキュリティは高まる。

##### 【図面の簡単な説明】

#### 【図1】

本発明の実施形態に係るセキュアLSIの構成を示すブロック図である。

#### 【図2】

図1のセキュアLSIを用いた開発および製品化の全体の流れを表す図である。

。

**【図 3】**

ブートプログラムの全体的な処理の流れを示すフローチャートである。

**【図 4】**

セキュアメモリ初期値設定 S Z 1 のデータフローである。

**【図 5】**

商品動作モードにおけるプログラム実装処理 S D 1 のフローチャートである。

**【図 6】**

プログラム実装処理 S D 1 のデータフロー 1 である。

**【図 7】**

プログラム実装処理 S D 1 のデータフロー 2 である。

**【図 8】**

商品動作モードにおける通常ブート処理 S D 2 のフローチャートである。

**【図 9】**

通常ブート処理 S D 2 のデータフロー 1 である。

**【図 1 0】**

通常ブート処理 S D 2 のデータフロー 1 である。

**【図 1 1】**

プログラム更新におけるサーバとの通信を示すフローチャートである。

**【図 1 2】**

外部メモリ 1 0 0 に格納された、プログラム更新に係わるプログラムの構成を示す図である。

**【図 1 3】**

プログラムの更新処理を示すフローチャートである。

**【符号の説明】**

- 1 セキュア L S I
- 3 サーバ
- 4 第 1 のテーブル
- 5 第 2 のテーブル
- 1 0 セキュアメモリ（内部メモリ）

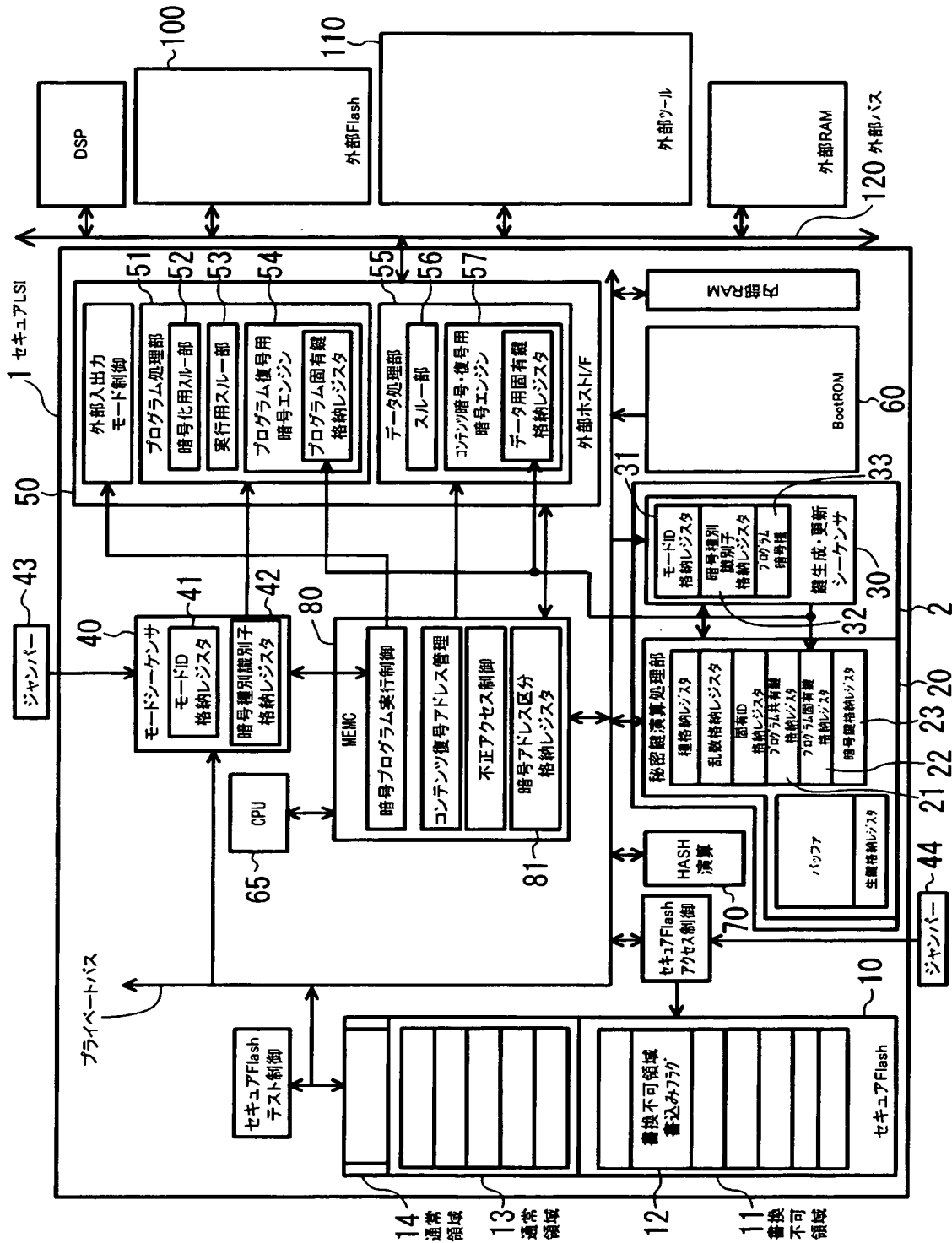
5 8 プログラム固有鍵格納レジスタ

6 0 ブートROM

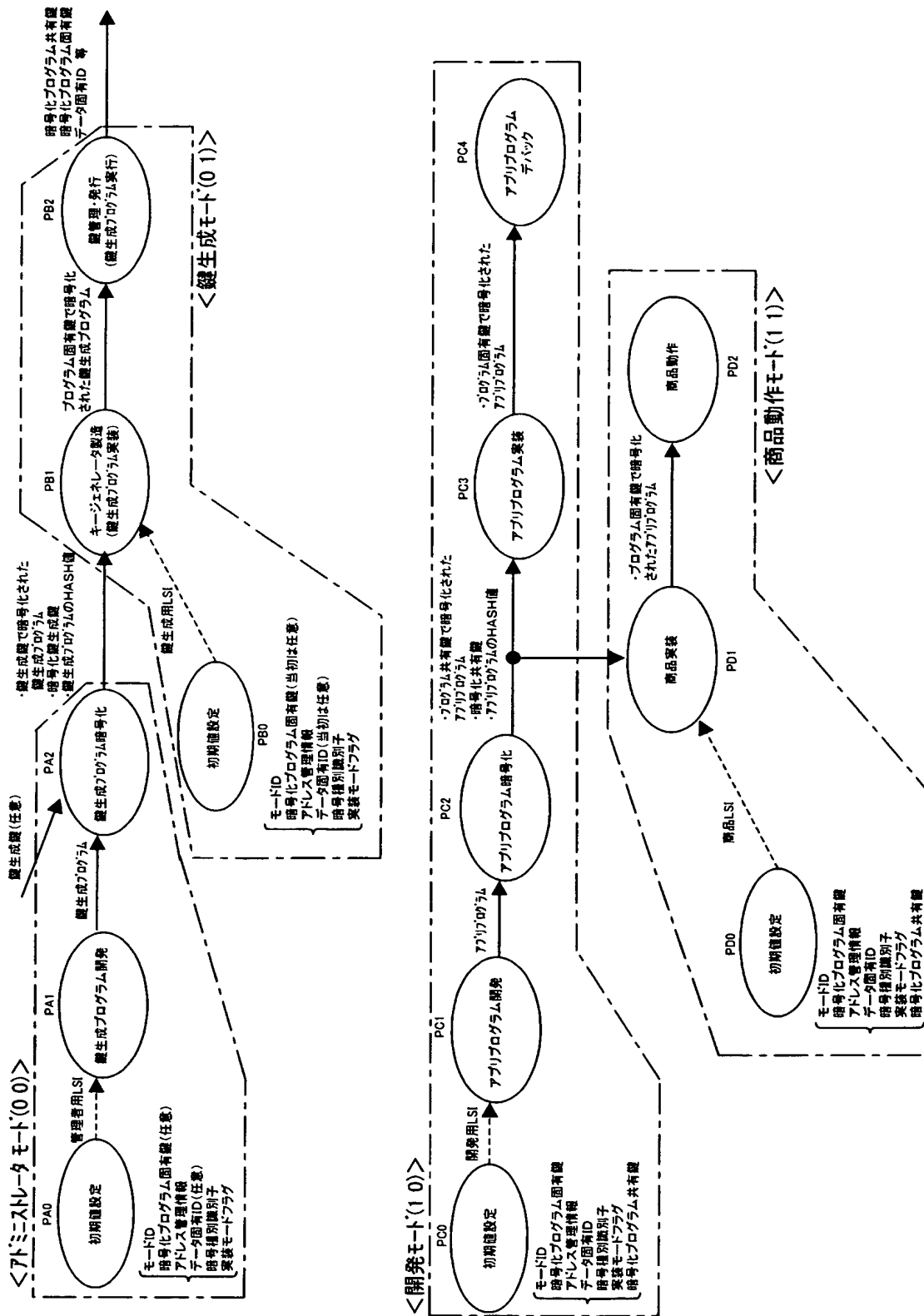
1 0 0 外部メモリ

【書類名】 図面

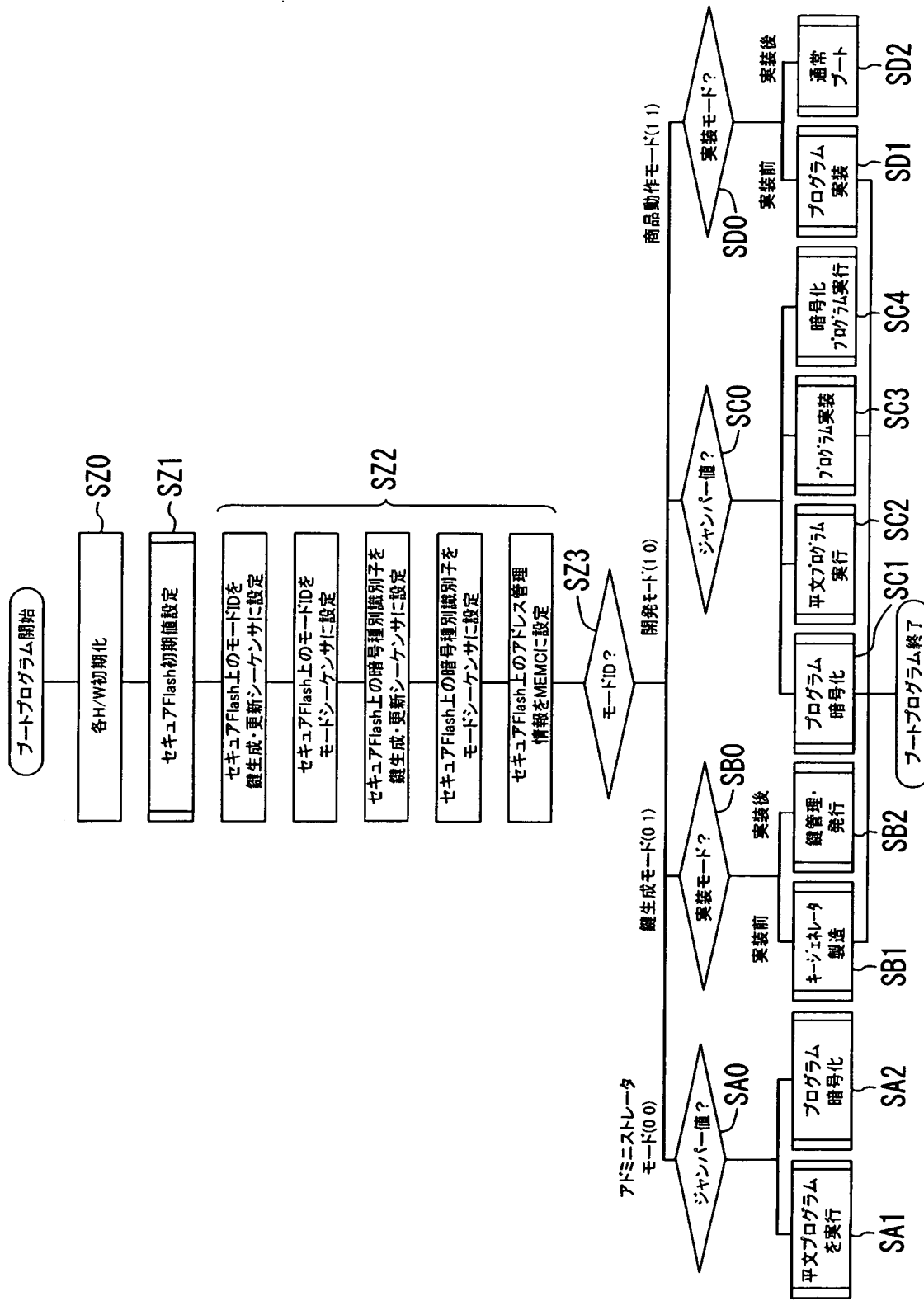
【図 1】



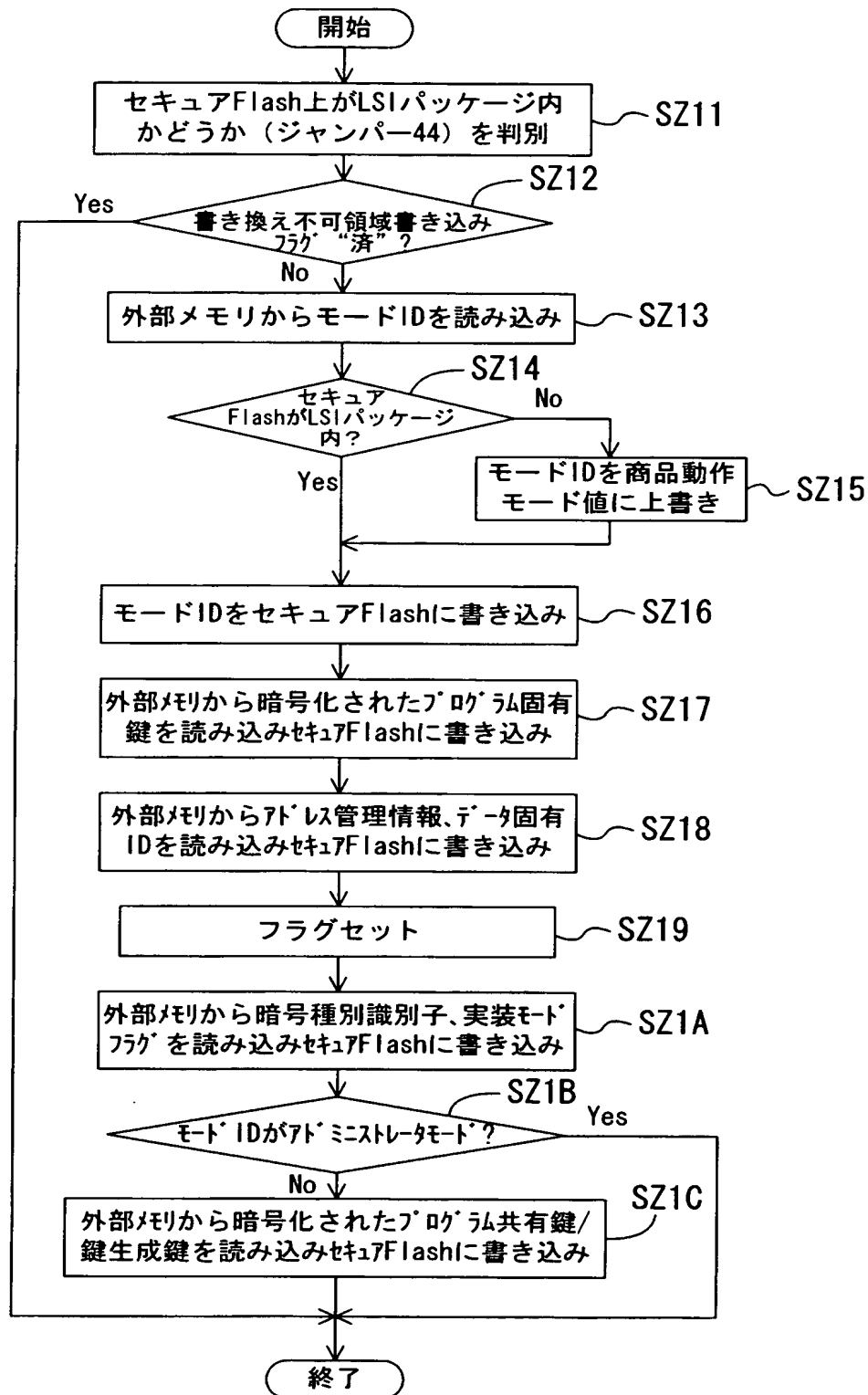
【図 2】



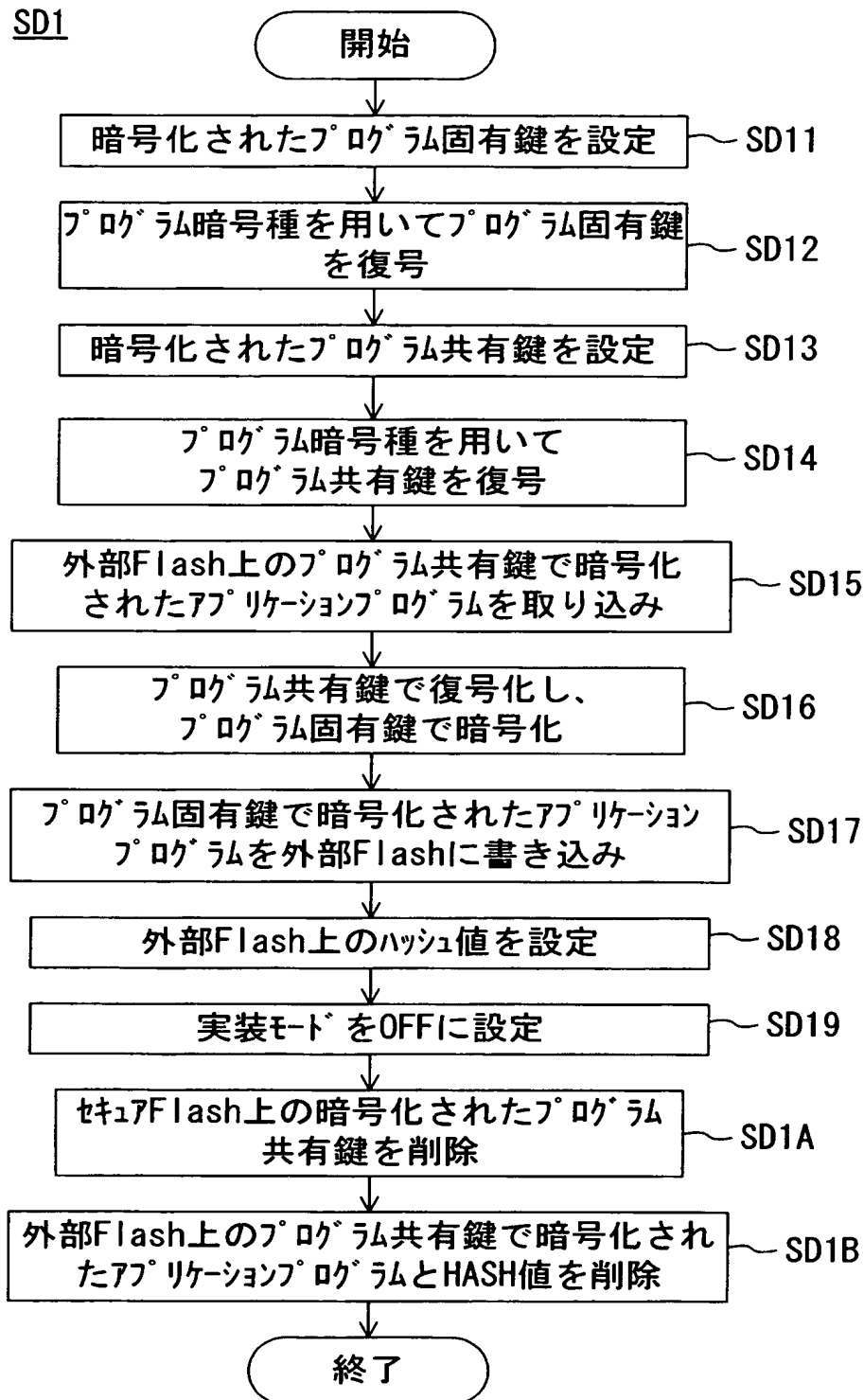
【図 3】



【図 4】

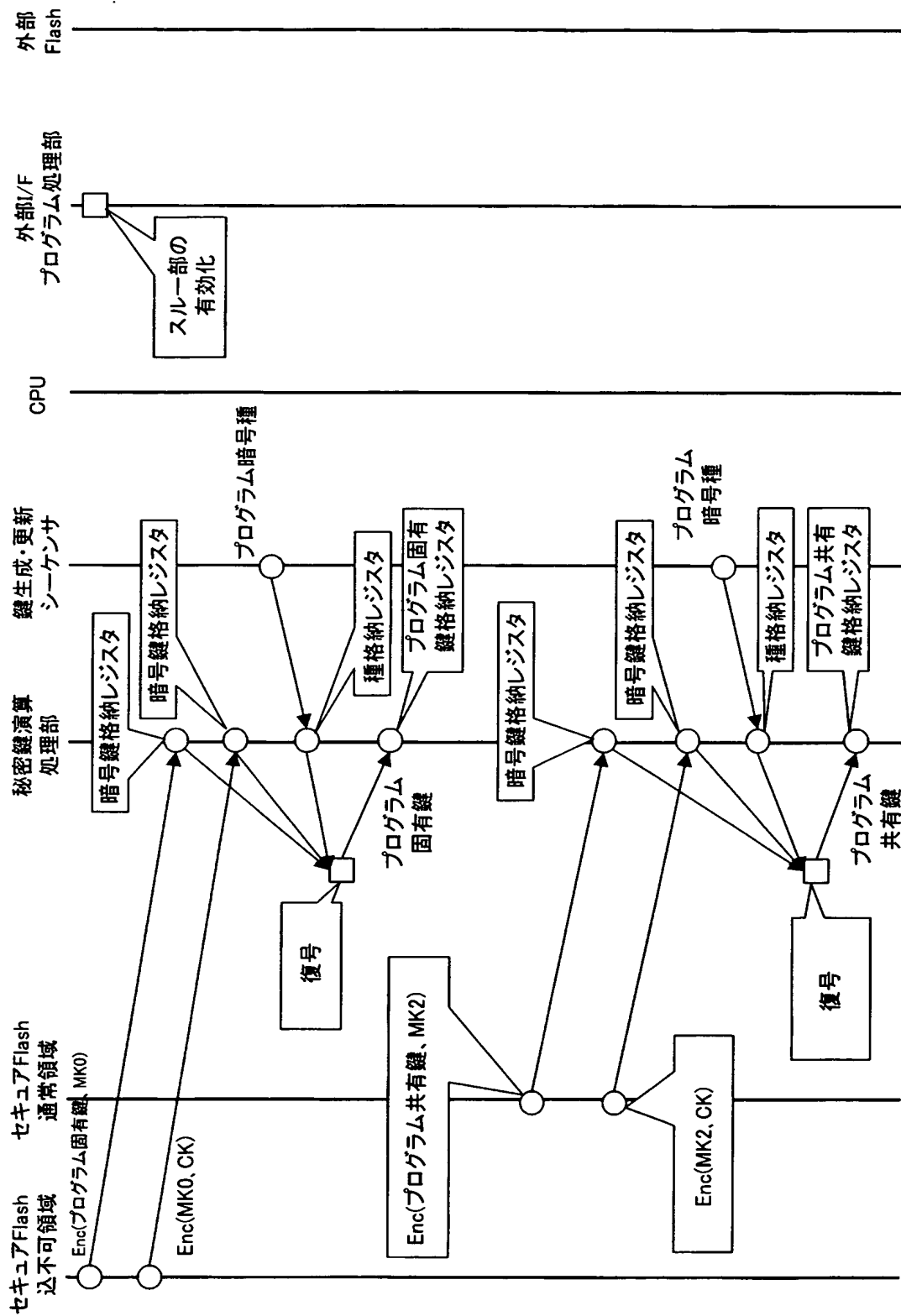
SZ1

【図 5】

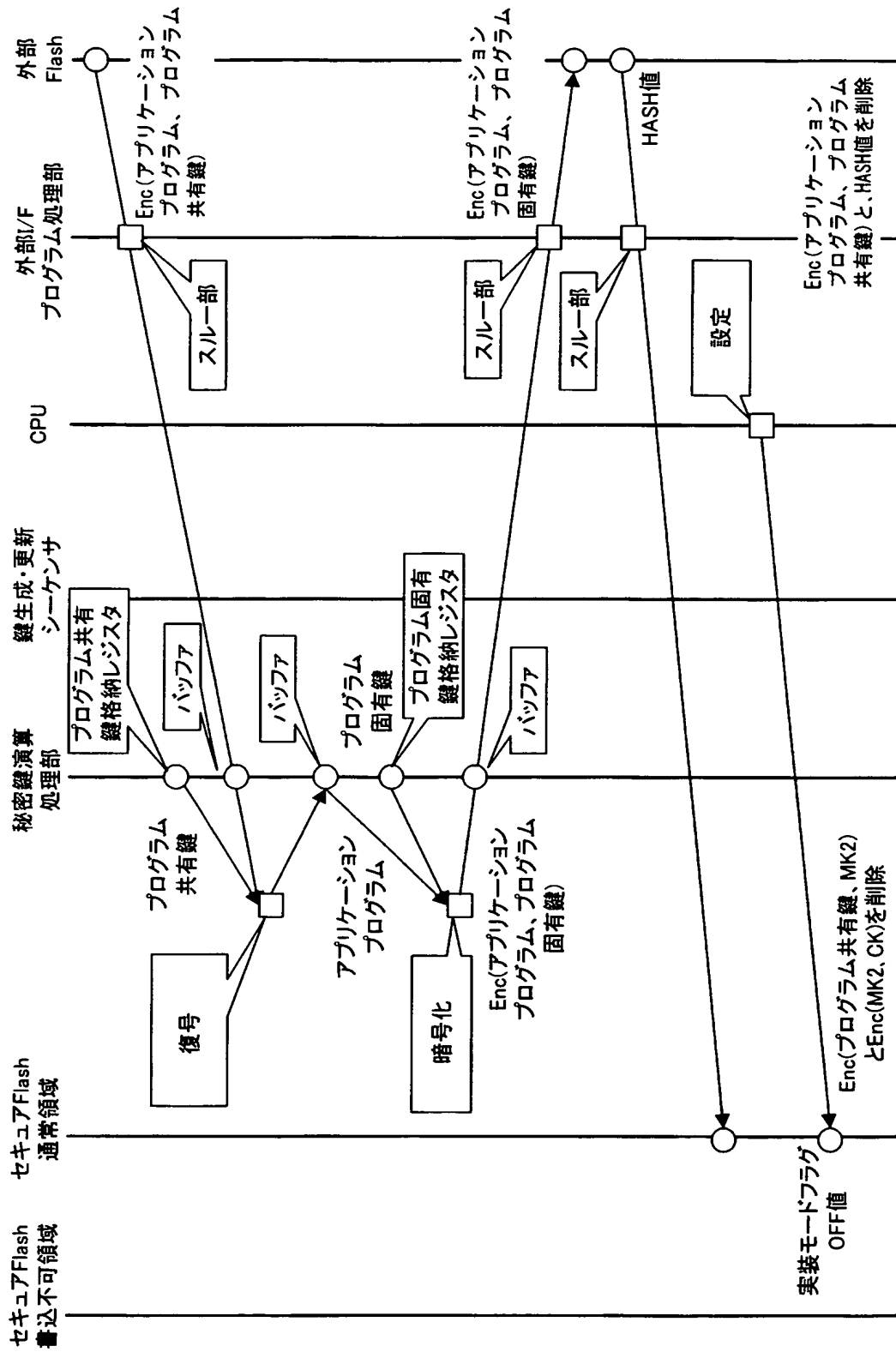




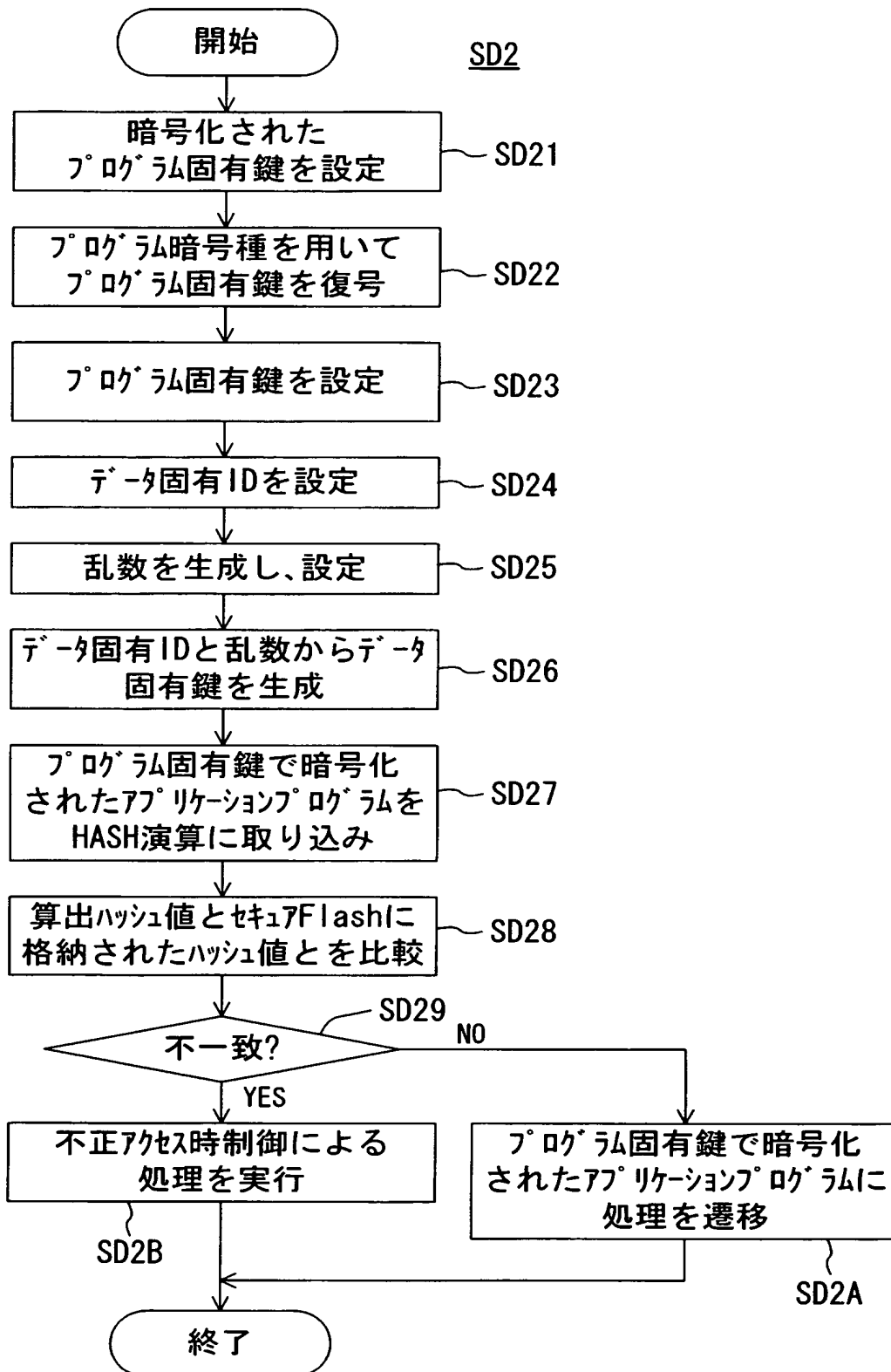
【図 6】



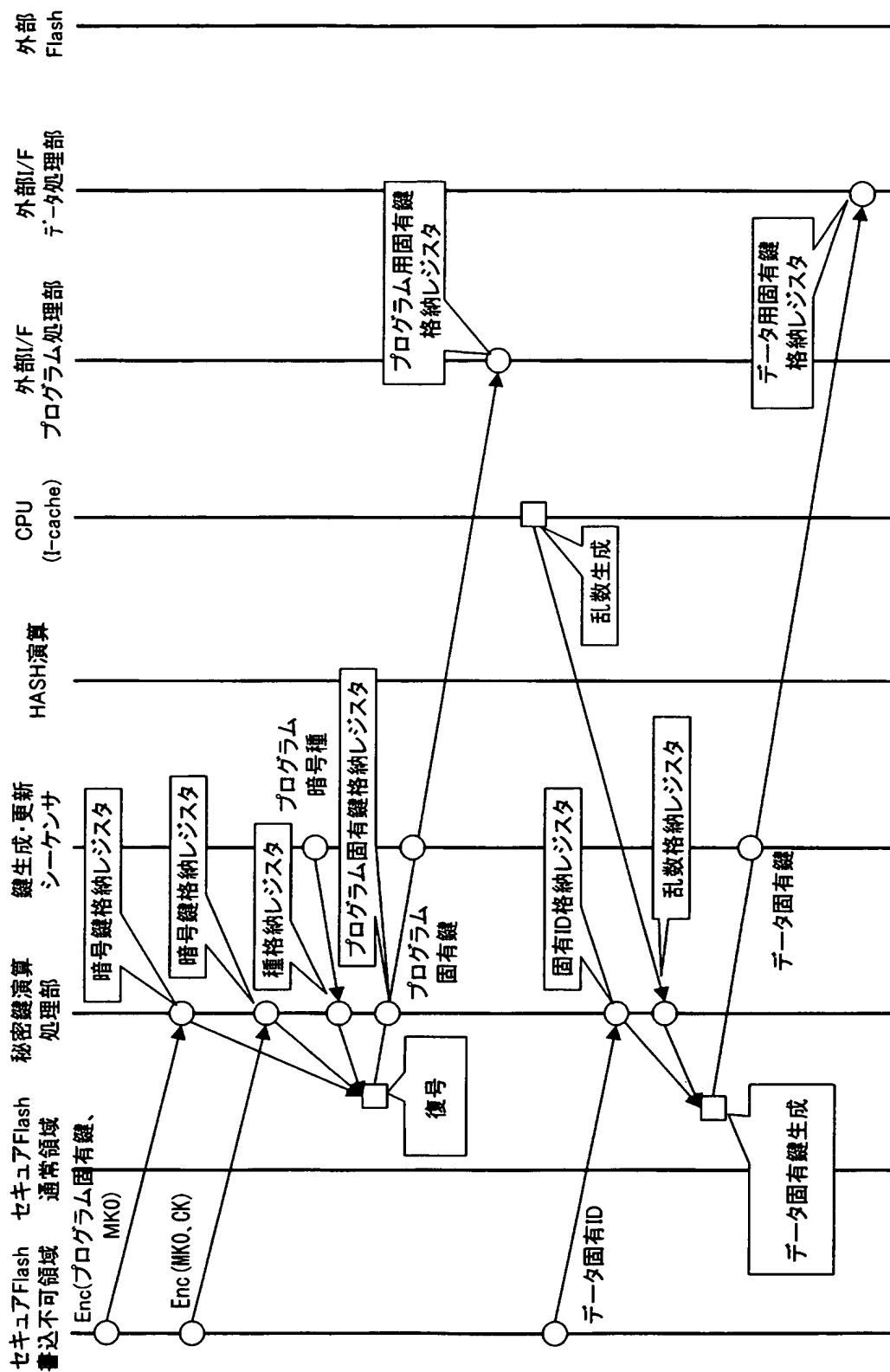
【図 7】



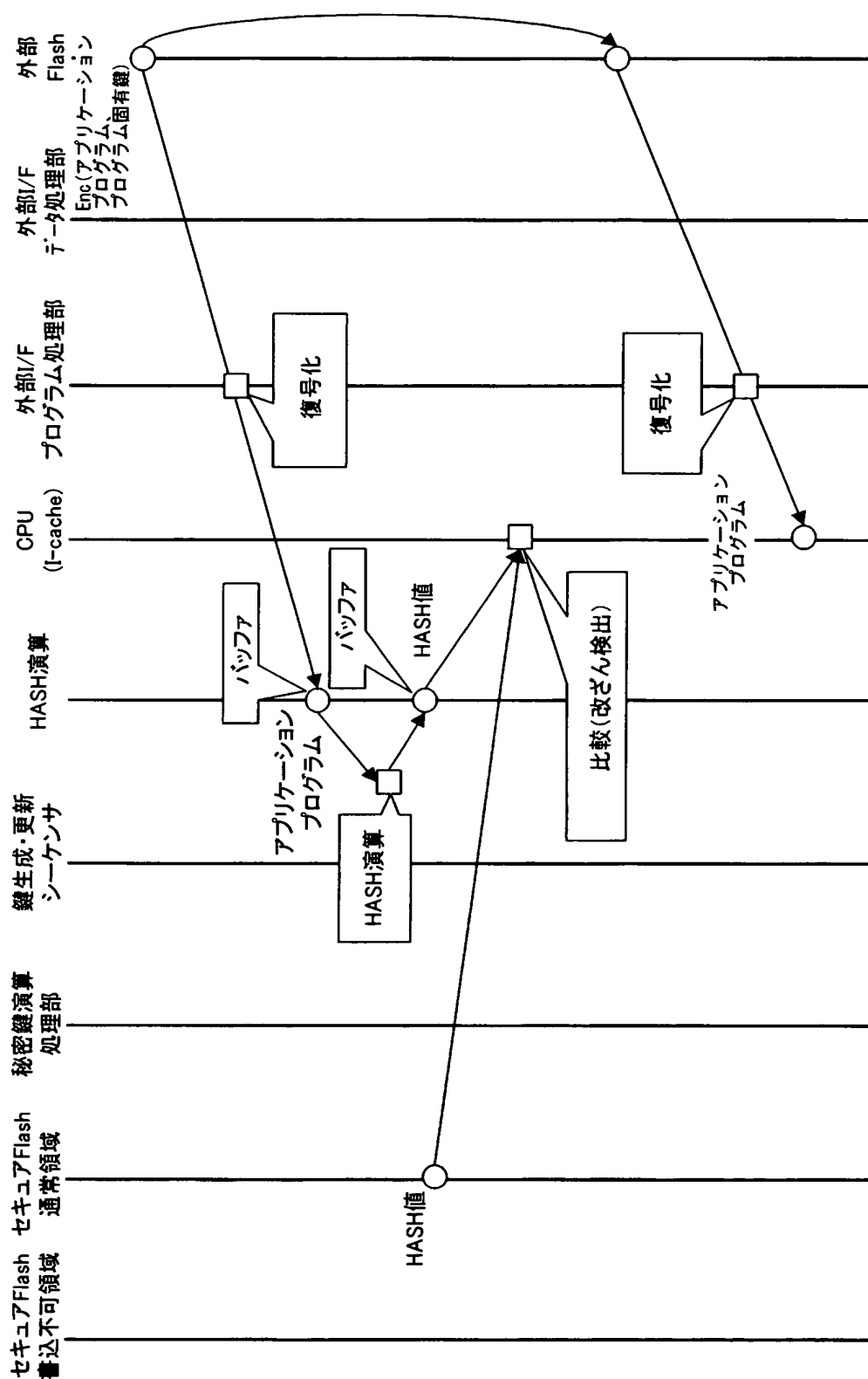
【図 8】



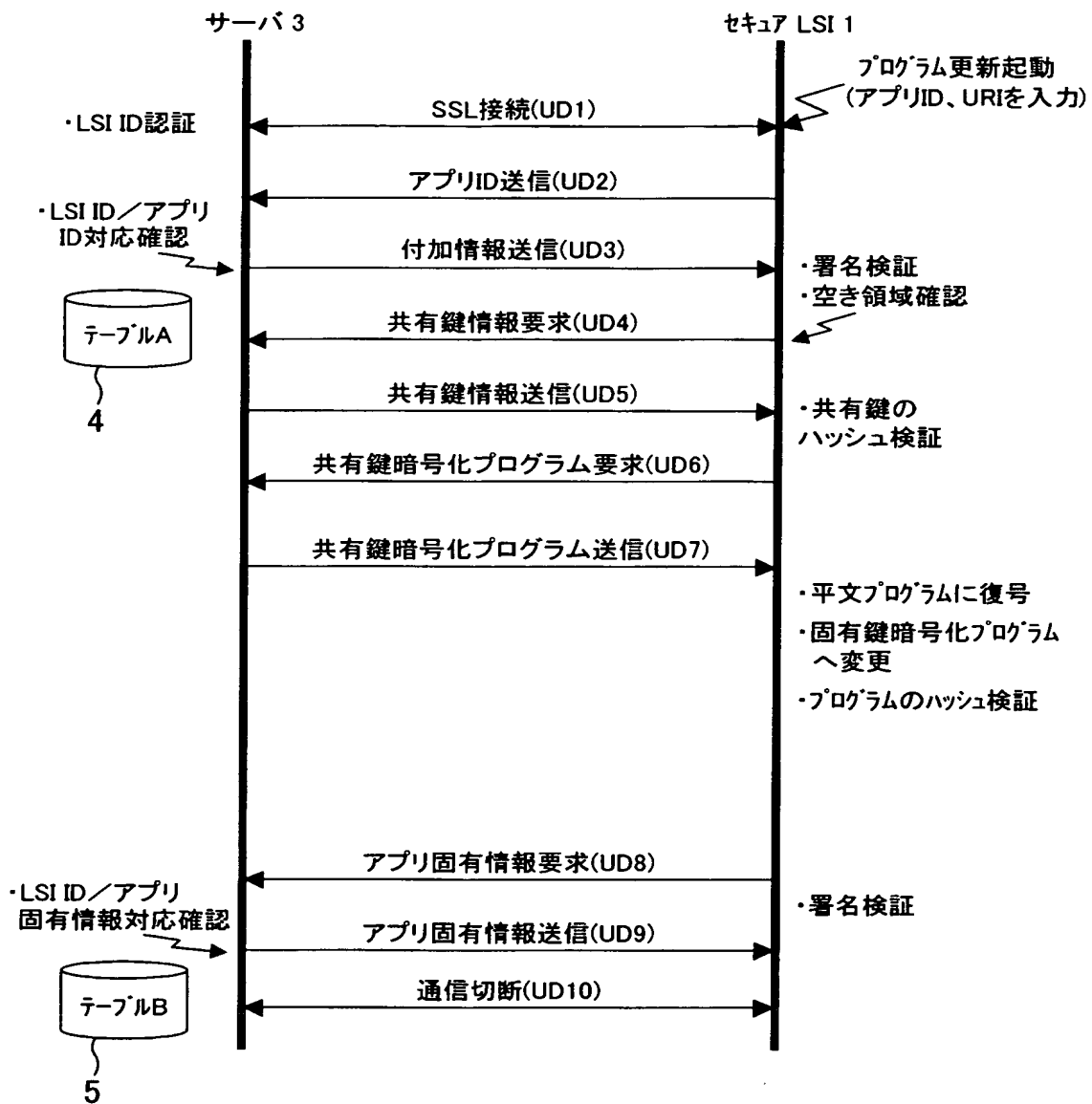
【図 9】



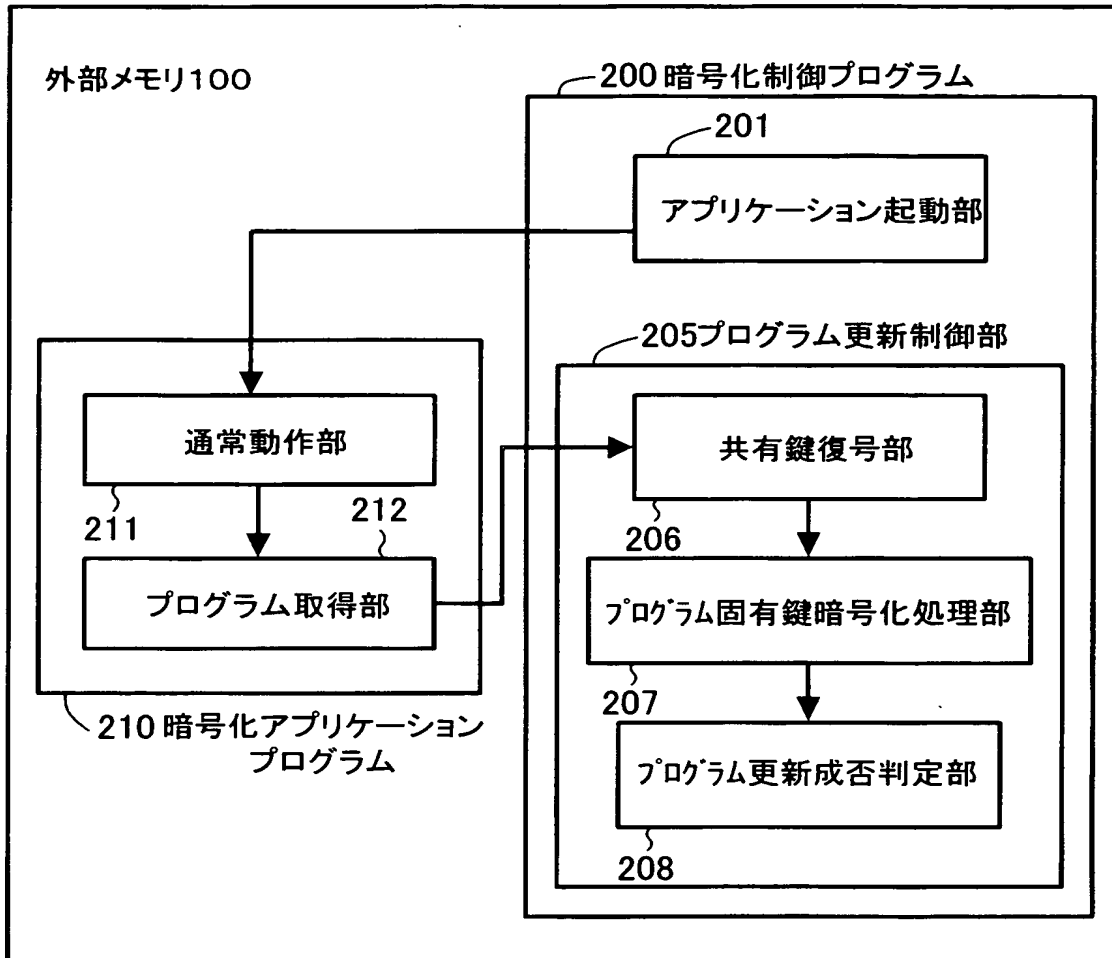
【図 10】



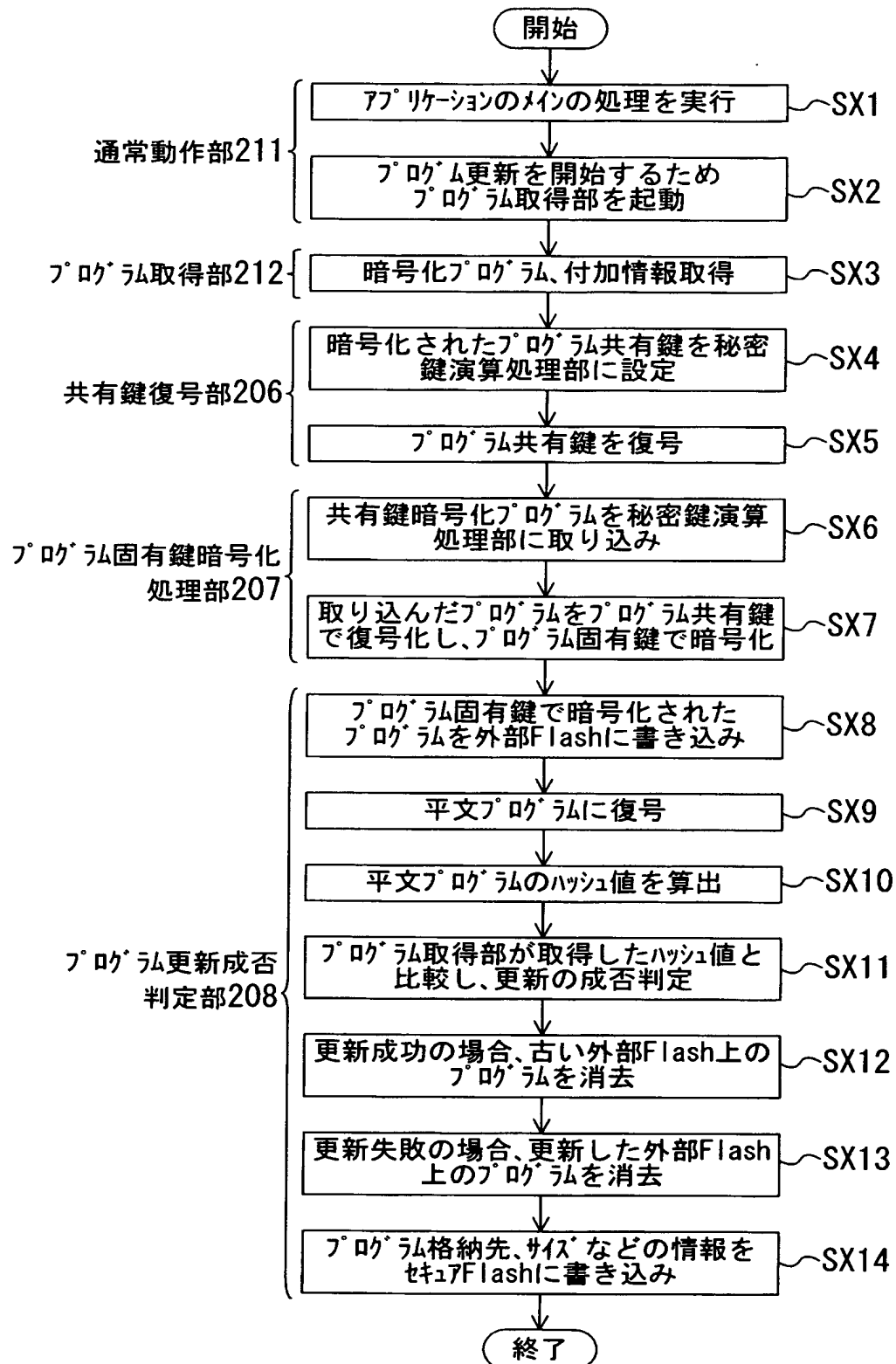
【図 11】



【図 12】



【図 13】





【書類名】 要約書

【要約】

【課題】 固有鍵で暗号化されたプログラムを実行可能な L S I について、高いセキュリティを保ちつつ、プログラムを更新可能にする。

【解決手段】 セキュア L S I 1 を含むシステムは、サーバ 3 との通信路を確立し（U D 1）、サーバ 3 から送信された、共有鍵で暗号化された共有鍵暗号化プログラムを受信する（U D 6，U D 7）。そして、受信した共有鍵暗号化プログラムを復号して平文プログラムを生成し、さらに、平文プログラムを固有鍵で再暗号化し、新たな固有鍵暗号化プログラムとして外部メモリに格納する。

【選択図】 図 1 1

特願 2 0 0 2 - 3 3 1 9 9 2

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 8 2 1 ]

1 . 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社